

**In the Drawings:**

A proposed change to Figure 3 marked in red ink is submitted herewith accompanied by a "Drawing Transmittal Letter". Upon approval of the proposed change, a formal drawing for Figure 3 will be submitted in due course.

**In the Claims:**

Please cancel claim 2.

Please amend claims 1, 3-5, 8-9, and 12-16 to read as follows:

AD Sub B1

1. (Amended) A method for processing software instructions comprising, decomposing a macroinstruction into a plurality of microinstructions, issuing all of the plurality of microinstructions simultaneously, in parallel, executing the at least two of the plurality of microinstructions simultaneously, in lockstep, determining whether an exception occurs in any of the microinstructions, and if an exception occurs in any of the microinstructions, canceling all of the microinstructions.

Q1

3. (Amended) The method of claim 1, wherein the microinstructions are executed on separate execution units, but appear as though they were executed on a single execution unit.

4. (Amended) The method of claim 1, wherein all of the microinstructions are executed on the same clock cycle.

5. (Amended) The method of claim 1, wherein the microinstructions are executed over multiple clock cycles.

At C1

8. (Amended) The method of claim 1, further comprising updating a flag based upon a result of the execution of the microinstructions.

9. (Amended) The method of claim 1, further comprising, if an unmasked exception occurs, canceling the execution of all of the plurality of microinstructions, without regard to the relative ages of each of the plurality of microinstructions, and invoking a microcode handler, and if an unmasked exception does not occur, updating at least one exception flag by independently generating a logical OR of exceptions for a plurality of functional units.

Sub B3

12. (Amended) A computer system comprising, a processor comprising, (a) a floating point unit;

Cont  
B3

(b) a ROM;

(c) a plurality of floating point registers;

wherein the processor is configured to emulate an instruction set by:

(a) decomposing a macroinstruction into a plurality of microinstructions;

(b) issuing all of the plurality of microinstructions simultaneously, in parallel,

(c) determining whether an exception occurs in any of the microinstructions, and

(d) if an exception occurs in any of the microinstructions, canceling all of the

microinstructions.

13. (Amended) The computer system of claim 12, wherein the processor is further configured to emulate the instruction set by executing all of the microinstructions.

14. (Amended) The computer system of claim 13, wherein the microinstructions are executed on separate execution units, but appear as though they were executed on a single execution unit.

15. (Amended) The computer system of claim 14, wherein the processor is further configured to emulate an instruction set by updating a flag based upon a result of the execution of the microinstructions.

16. (Amended) The computer system of claim 15, wherein the processor is further configured to emulate an instruction set by

determining whether an exception occurs in the execution of any of the microinstructions,

if an exception occurs, causing the exception to cancel all of the microinstructions.

Please add the following new claims 19-20:

19. (New) The method of claim 1, wherein the step of issuing comprises forcing the microinstructions to issue simultaneously, in lockstep with each other, and wherein the step of canceling comprises canceling all of the plurality of microinstructions without regard to the relative ages of the microinstructions and without using a backoff mechanism.

20. (New) The method of claim 10, wherein the step of forcing the high-half and low-half operations to issue in parallel comprises causing the high-half and low-half operations to execute simultaneously in lockstep with each other, and wherein the step of flushing a result comprises canceling each of the high-half and low-half operations if an exception is taken in either the first or second FP unit.